Computer Science and Information Technologies
Vol. 5, No. 3, November 2024, pp. 227~234
ISSN: 2722-3221, DOI: 10.11591/csit.v5i3.p227-234

Securing DNS over HTTPS traffic: a real-time analysis tool

Abid Dhiya Eddine, Ghazli Abdelkader
Department Mathematics and Computer Sciences, Faculty of Exact Sciences, Tahri Mohamed University of Bechar, Algeria

Article Info

Article history:

Received Apr 19, 2024
Revised Jul 27, 2024
Accepted Jul 31, 2024

Keywords:

Artificial intelligence
Cybersecurity
Deep learning
Domain Name System
Hypertext transfer protocol secure
Machine learning
Threats detection

ABSTRACT

DNS over HTTPS (DoH) is a developing protocol that uses encryption to secure domain name system (DNS) queries within hypertext transfer protocol secure (HTTPS) connections, thereby improving privacy and security while browsing the web. This study involved the development of a live tool that captures and analyzes DoH traffic in order to classify it as either benign or malicious. We employed machine learning (ML) algorithms such as K-nearest neighbors (K-NN), random forest (RF), decision tree (DT), deep neural network (DNN), and support vector machine (SVM) to categorize the data. All of the algorithms, namely KNN, RF, and DT, achieved exceptional performance, with F1 scores of 1.0 or above for both precision and recall. The SVM and DNN both achieved exceptionally high scores, with only slight differences in accuracy. This tool employs a voting mechanism to arrive at a definitive classification decision. By integrating with the Mallory tool, it becomes possible to locally resolve DNS, which in turn allows for more accurate simulation of DoH queries. The evaluation results clearly indicate outstanding performance, confirming the tool's effectiveness in analyzing DoH traffic for network security and threat detection purposes.

Corresponding Author:

Abid Dhiya Eddine
Department of Mathematics and Computer Sciences, Faculty of Exact Sciences
Tahri Mohamed University of Bechar
Istiklal Street, Bechar, 08000, Algeria
Email: abid.dhiyaeddine@gmail.com

1. INTRODUCTION

The Internet has completely transformed how we interact with the outside world, conduct business, communicate, and obtain information. It is a vast, globally connected computer network with servers spread across the globe, enabling the seamless exchange of data and digital content. From its humble beginnings as a research project to its ubiquitous presence, the Internet has become an integral part of modern society, forming how we live, work, and connect [1].

Despite all of this, our Internet and technology use remain highly vulnerable to many threats that constantly put our data at risk of theft or hacking. This leads us to search for ways to make our data safer by developing effective methods to detect and eliminate all these threats. Undoubtedly, most of these methods today originate from artificial intelligence.

Artificial intelligence (AI) has become a potent instrument for identifying potential threats and mitigating them. AI techniques like machine learning (ML) and deep learning (DL) examine enormous volumes of data to find trends, abnormalities, and possible dangers in digital systems. These advanced algorithms can detect and predict cyber attacks ranging from malware and phishing attempts to network intrusions and data breaches. By leveraging AI, organizations can bolster their cyber security defenses and stay

ahead of cyber threats, enhancing their overall resilience in an increasingly complex and sophisticated digital environment [2].

One of the biggest issues facing the world and companies today is privacy on the Internet. This has led to a growing interest in developing new technologies that preserve user privacy while using the Internet and prevent unauthorized parties from accessing their data. One such technology is the so-called DoH or DNS over HTTPS.

The DoH technique secures communication between devices and DNS servers [3]. When a device requests a site, it sends the request to the domain name system (DNS) server for an IP address. However, the response is unencrypted, allowing any party to intercept, modify, and redirect the computer to other harmful sites. This vulnerability is illustrated in Figure 1, which shows how an attacker can intercept and modify an unencrypted DNS request, leading to the user being redirected to a malicious site. To mitigate the risk, the hypertext transfer protocol secure (HTTPS) protocol encrypts and completely secures the response, preventing any party, including its service provider, from accessing user data. Figure 2 demonstrates a scenario where DoH is active, ensuring that the DNS request is securely encrypted, and the user receives the correct response, even if an attacker tries to intercept the request.

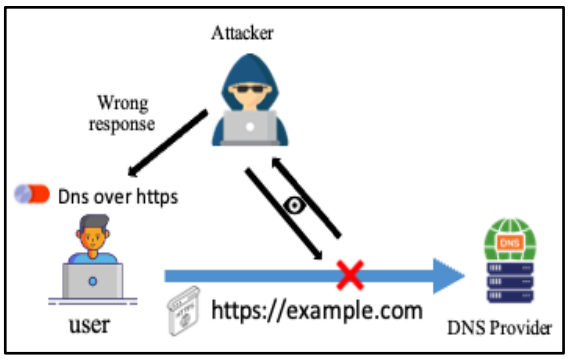

Figure 1. Illustration of a scenario where DoH is not active

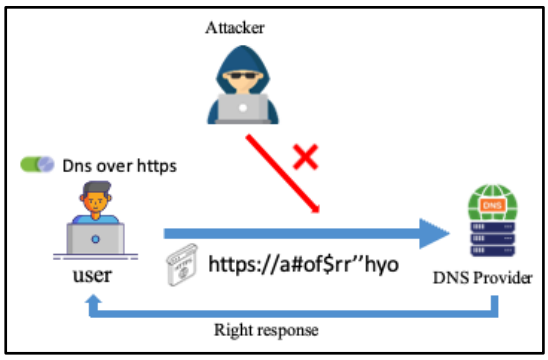

Figure 2. Illustration of a scenario where DoH is active

DoH is increasingly important to protect DNS messages from third-party attacks [4]. DNS over TLS and DoH encryption conceal DNS resolvers from passive adversaries [5]. Although DoH reduces the risk of data breaches, managing DNS traffic becomes more challenging for network security services due to its encryption [6].

The most recent attempts to secure DNS using DoT and DoH are gaining traction to protect traffic and hide content from unauthorized viewers [4]. However, DoT and DoH can only cover the interaction between the final client and the DNS full-service resolver, so they cannot defend against cache-destroying attacks [7]. While DoH provides Internet users with desirable features like privacy and security, it also makes it difficult for network managers to identify questionable network traffic produced by malicious software and viruses [8]. Protocols such as FTP-DNS, HTTP-DNS, HTTPS-DNS, and POP3-DNS can implement DNS tunneling [9].

Studies like [10]–[12] applied ML algorithms to create models that can detect and classify malicious DoH traffic; however, studies like [13] applied a DL approach to track the DoH network traffic. The [14] work also utilized DL techniques to evaluate and enhance learning-based DoH traffic. The paper [15] used a two-layered approach and time-series classification of encrypted traffic to detect DoH tunnels, and they called for ML and DL techniques. Similarly to the previous work, [16] proposes a two-stage and lightweight approach to detecting malicious DoH traffic using random fine trees in the first layer and AdaBoost trees in the second layer [17]. Developed a hierarchical ML classification method to pinpoint malicious DoH. In order to detect DoH attacks, [8] implemented an explainable AI solution using balanced and stacked random forests (RF).

On the other hand, this study aims to enhance the analysis of DoH in network traffic by combining two methods: ML and DL. Crucially, we will incorporate these models into a tool we have developed to capture packets in real-time and categorize the network traffic.

In this paper, we will discuss the development of a tool that can track network traffic in real time, test DoH, and classify malicious and benign traffic based on several DL and ML models integrated into it. In the training process, we will use the public dataset CIRA-CIC-DoHBrw-2020 to build our classification models. Next, we will develop a tool for real-time packet sniffing, feed it into our models, and then utilize a

voting technique to categorize the traffic. To prove our tool's effectiveness, we will simulate the creation of malicious requests on our local device so that our tool can recognize them and classify them as malicious.

2. METHOD

The following methodological steps were undertaken to create a tool that can classify DoH traffic in real-time as benign or malicious. Before diving into the details, Figure 3 briefly illustrates our research methodology. Which involves multiple stages from the dataset preprocessing to final tool deployment:

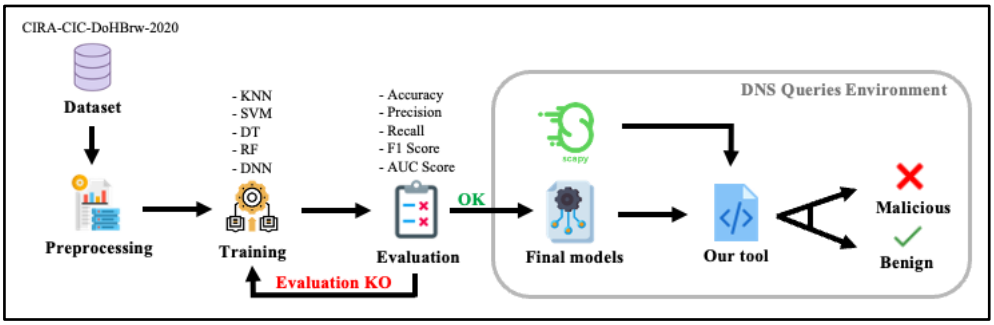

Figure 3. The proposed methodology

2.1. Data training

This study calls for Python programming language, scikit-learn, TensorFlow, and Keras libraries and uses the CIRA-CIC-DoHBrw-2020 dataset for analysis. Initially, 20 important features were chosen from 32 features from the dataset to represent different characteristics of DoH traffic, including source and destination ports, duration, and flow byte sent. To ensure the effectiveness of the training process, we first examined the correlation between the features. High correlation between features can negatively impact the performance of ML models. Therefore, we removed features with a high correlation threshold of 0.7. Figure 4 illustrates the heatmap of feature correlations before the removal of highly correlated features. As shown, several features exhibit high correlation, indicated by the dark green squares. After identifying these highly correlated features, we proceeded to remove them to reduce redundancy and improve model performance. Figure 5 shows the heatmap after removing the highly correlated features. The reduction in high correlation instances demonstrates a more refined feature set for training our models.

The preprocessing step, including scaling, was then applied to prepare the data for modeling. Subsequently, classification models, including K-nearest neighbors (KNN), RF, support vector machine (SVM), decision tree (DT), and deep neural network (DNN), were trained on the preprocessed features.

To assess the performance of the classification system, evaluation metrics were used, including accuracy, which is the percentage of cases accurately classified out, it calculated by the following in (1):

$$Accuracy = \frac{(TP+TN)}{TP+TN+FP+FN} \quad (1)$$

Precision which is the measure of the accuracy of positive predictions, it calculated by the following in (2):

$$Precision = \frac{\text{TP}}{TP+FP} \quad (2)$$

Recall which is the percentage of actual positive cases divided by the true positive forecasts, it calculated by the following in (3):

$$Recall = \frac{\text{TP}}{TP+FN} \quad (3)$$

The F1-score is the recall and precision harmonic means which strikes a harmony between the two measures [18], it calculated by the following in (4):

$$F1\ Score = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision+Recall} \quad (4)$$

Another method for evaluating the performance of classification models is called receiver operating characteristic (ROC), and It is a graphical representation that shows how a binary classifier system may be made to function as a diagnostic across different threshold settings. The ROC curve illustrates the trade-off between the classifier's sensitivity and specificity by plotting the true positive rate (TPR) versus the false positive rate (FPR) at different threshold levels. AUC is a scale from 0 to 1, with a higher number denoting the model's superior discriminating power [19].

The ML models including KNN, SVM, DT and RF are trained using scikit-learn library how ever the DNN trained using TensorFlow and Keras libraries. Scikit-learn is a quick and easy tool for analyzing predictive data [20] and Tensorflow is a ML end-to-end platform [21].

After the training process is completed and good results are obtained, a model with the h5 extension is extracted to be used while the tool is running.

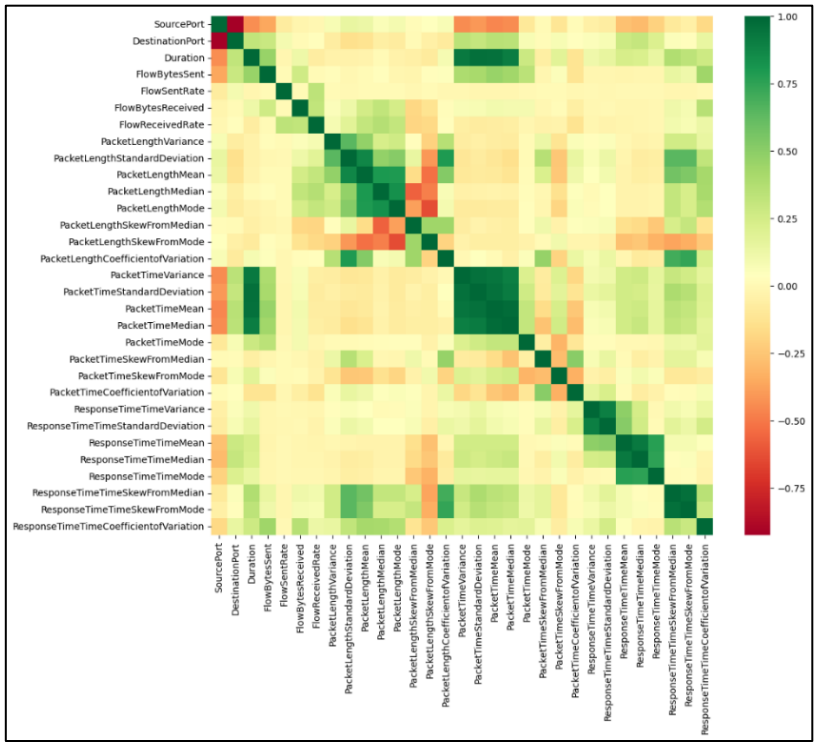

Figure 4. Heatmap before removing correlation

2.2. Tool development

In the development step, we used Python programming language and Scapy library to create our real-time tool. Scapy is a robust Python library for manipulating packets interactively. Scapy can transmit packets across the network, capture them, match requests and answers, and do a lot more. It can also forge or decode packets of a variety of protocols [22].

Our tool sniffs the packets and extracts some features from them, such as source port, source IP, destination IP, and destination port. Some other features are calculated using statistical rules, such as packet length variance and mode. All the extracted features will pass to each model to make its prediction (voting), and then the most voted decision, benign or malicious, will be chosen. The following Algorithm 1 resumes how our tool will work.

2.3. Simulation

In this section, we used the well-known Man-in-the-Middle (MitM) proxy, Mallory, to imitate malicious DoH traffic. Mallory is a tool used by researchers to examine network activity and possible vulnerabilities since it can intercept and alter HTTPS traffic, including DoH requests and responses [23]. Using this tool, we were able to improve the robustness of our research by generating realistic situations that mimicked possible vulnerabilities inside DoH traffic.

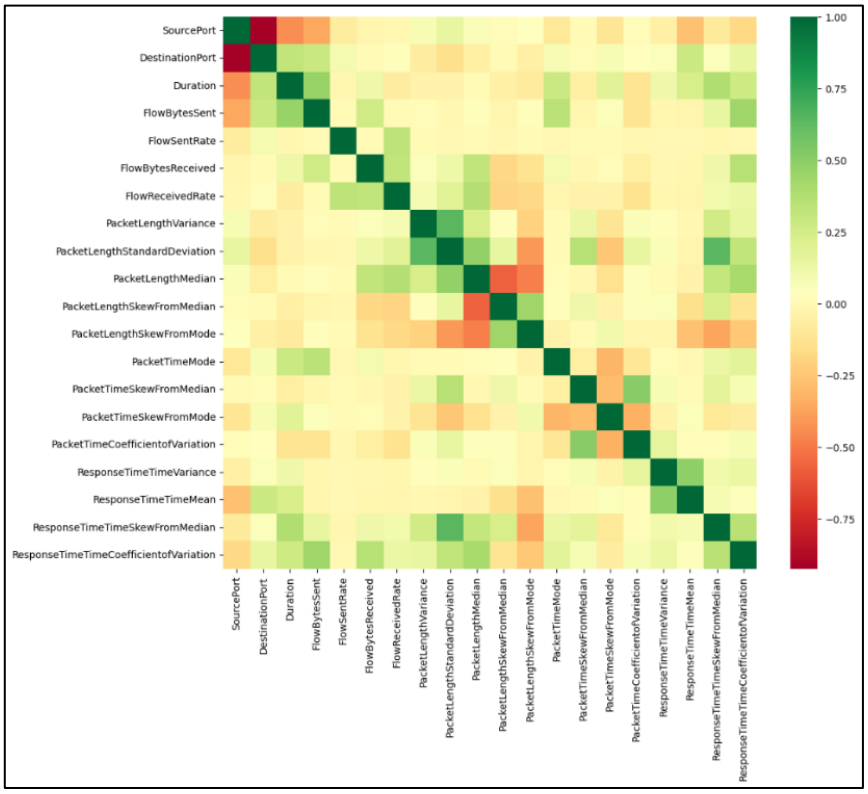

Figure 5. Heatmap after removing correlation

Algorithm 1: Sniff and classify

```
    Input: vector of features such as source port, destination port, duration, and
    packet length.
    Output: decision benign or malicious
1   knn_classifier, svm_classifier, dt_classifier, rf_classifier, dnn_classifier ← load_models()
2   While (true) do
3     packet ← scapy.sniff()
4     features ← packet.extract_features()
5     knn ← knn_classifier.predict(features)
6     svm ← svm_classifier.predict(features)
7     dt ← dt_classifier.predict(features)
8     rf ← rf_classifier.predict(features)
9     dnn ← dnn_classifier.predict(features)
10    decision ← max_vote_between(knn, svm, dt, rf, dnn)
11    if (decision = 'malicious') then
12      show_malicious_doh_data(features)
13    End
14  End
```

3. RESULTS AND DISCUSSION

This section presents some results of our research with a final discussion, where the aim is to create a real-time tool for analyzing DoH traffic.

3.1. Results

The first step of our work was training the CIRA-CIC-DoHBrw-2020 dataset; in this step, we used four algorithms in the ML approach, which are KNN, SVM, DT, and RF. However, we used the DNN for the DL approach. Table 1 summarizes the training results for these models. We used accuracy, precision, recall, F1 score, and AUC score as the evaluation metrics to assess the performance of our final models. As mentioned above, the AUC score also used to evaluate our models. The AUC score represents the scalar value that quantifies the overall performance of the classifier [24], [25].

After training the dataset and extracting all the models, we succeeded in the development part, where we developed our tool, which sniffs packets using the Scapy library and then passes each packet to all models in order to make their prediction; after that, the process will continue by counting the number of predictions in order to get the final decision (voting). After the development task, we configured the Mallory tool in our local machine in order to simulate malicious DoH queries, and we got good results where our tool was able to analyze and classify the queries successfully.

Table 1. Training results

	Precision	Recall	F1 Score	Accuracy	AUC Score
KNN	1	1	1	1	0.9971
SVM	0.99	0.99	0.99	0.99	0.9922
DT	1	1	1	1	0.9968
RF	1	1	1	1	0.9983
DNN	0.96	0.99	0.97	0.99	0.9875

3.2. Discussion

In this study, we created a tool for analyzing DoH in real time; we started by exploring the efficacy of various ML models, including traditional classifiers (KNN, RF, DT, SVM) and a DNN, in analyzing DoH traffic. Our analysis included a dataset with 30 features, from which 20 were selected after removing highly correlated ones with a threshold of 0.7. As a performance discussion for our training results we observed exceptionally high-performance metrics, with precision, recall, and F1 score reaching 1.0 for KNN, RF, and DT, and perfect scores for SVM and DNN. Notably, SVM and DNN showed a few minor differences in performance metrics, achieving an accuracy of 0.99 compared to the perfect accuracy of the other models. These results underscore the robustness of our models in accurately classifying DoH traffic. After the success of training, we developed our tool using Python and its libraries including scapy, scikit-learn and TensorFlow. Then, we simulated some malicious DNS queries using Mallory, and our tool succeeded in the detection of these malicious queries.

As a simple comparison with similar works mentioned in the introduction to this research, we find that perhaps the work [10], [13] better than ours in terms of choosing the dataset because they used their dataset collected at the level of their local servers. On the other hand, the research [16] was able to obtain good results using only six features from the same dataset that we used, but in the end, we all obtained satisfactory results in the training phase despite some differences in the training techniques and algorithms used. While our work excels in not stopping at creating trained models, we have also created a tool that uses these models in order to analyze and classify the DoH in real-time and in addition to ensuring Its effectiveness by carrying out the simulation process. Table 2 (see on appendix) provides a comparison of our research with existing works, detailing the datasets used, the training techniques applied, the number of features considered, and the evaluation metrics used. This comparison helps to contextualize our results within the broader landscape of related research.

4. CONCLUSION

Finally, by presenting a real-time tool strengthened with AI methodologies, our research represents a major improvement in the field of DoH investigation. We have proven the effectiveness of our technology in precisely identifying and analyzing encrypted DNS traffic by rigorously testing SVM, KNN, DT, RF, and DNN models. The entire suite of performance indicators, which includes F1 score, accuracy, precision, and recall, attests to the dependability and harmony of our methodology.

Furthermore, as demonstrated by Mallory simulations, our tool's robustness in realistically simulated environments highlights its usefulness for network security and monitoring. Our program is a major advancement in DoH analysis since it incorporates novel features including real-time packet sniffer using the Scapy library and addresses constraints found in previous research. Going forward, our tool needs to be integrated with the operating systems or with the browsers for enhacing the security of systems. In essence, our study not only contributes to the growing body of knowledge in network security but also holds promise for bolstering defenses against emerging threats posed by encrypted DNS traffic. As we continue to innovate and refine our approach, the potential for our tool to serve as a cornerstone in safeguarding network integrity and privacy remains ever-promising.

APPENDIX

Table 2. Comparaison with similar works

Ref	Dataset	Training techniques	Features Number	Evaluation Metrics	Implementation (model/app)
[3]	Custom	- KNN - C4.5 - RF - Naïve Bayes - Ada-boosted DT	19	Accuracy	Models
[4]	CIRA-CIC-DoHBrw-2020	- Decision Tree - Extra Tree - Gradient Boosting - LGBM - RF - XGBoost	34	- Confusion matrix - ROC - AUC - Accuracy - Precision - Recall - F1 score	Models
[5]	CIRA-CIC-DoHBrw-2020	- Logistic Regression - Random Forest - KNN - Gradient Boosting - Naïve Bayes	27	- Confusion matrix - Precision - Recall - F1 score	Models
[6]	Custom	- Convolutional neural network	90	Accuracy	Models
[7]	CIRA-CIC-DoHBrw-2020	- KNN - 1D CNN - 2D CNN - Long short-term memory	34	- Accuracy - Precision - Recall - F1 score	Models
[8]	CIRA-CIC-DoHBrw-2020	- RF - C4.5 - SVM - NB - DNN - 2D CNN	34	- Precision - Recall - F1 score	Models
[9]	CIRA-CIC-DoHBrw-2020	- Principal component analysis - Random fine trees - Adaboost trees	6	- Accuracy - Overhead	Model
[10]	CIRA-CIC-DoHBrw-2020	- XGBoost - LightGBM - CatBoost	More than 28	- Accuracy - Precision - Recall - F1 score - Mean Time Between False Alarms	Models
[11]	CIRA-CIC-DoHBrw-2020	- Balanced and stacked random forest - DT - GB - RF	29	- AUC - Accuracy - Precision - Recall - F1 score	Model
Our research	CIRA-CIC-DoHBrw-2020	- KNN - SVM - DT - RF - DNN	20	- AUC - Accuracy - Precision - Recall - F1 score	- Models - Real-time Tool

REFERENCES

[1] J. MCKir, "Inventing the Internet," *Canadian Journal of Communication*, vol. 26, no. 1, pp. 157–159, Jan. 2001, doi: 10.22230/cjc.2001v26n1a1202.

[2] B. M. Leiner *et al.*, "A brief history of the internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, Oct. 2009, doi: 10.1145/1629607.1629613.

[3] T. Böttger *et al.*, "An empirical study of the cost of DNS-over-HTTPS," in *Proceedings of the Internet Measurement Conference*, New York, NY, USA: ACM, Oct. 2019, pp. 15–21. doi: 10.1145/3355369.3355575.

[4] S. Singanamalla *et al.*, "Oblivious DoH (ODoH): a practical privacy enhancement to DNS," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 4, pp. 575–592, Oct. 2021, doi: 10.2478/popets-2021-0085.

[5] J. Bushart and C. Rossow, "Padding ain't enough: assessing the privacy guarantees of encrypted DNS," Jul. 2019, doi: 10.48550/arXiv.1907.01317.

[6] R. Mitsuhashi, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Malicious DNS tunnel tool recognition using persistent DoH traffic analysis," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 2086–2095, Jun. 2023, doi: 10.1109/TNSM.2022.3215681.

[7] S. M. Yong Jin, Masahiko Tomoishi, "Forged cache isolation on DNS full-service resolvers and identification of infected end clients," in *Proceedings of 2022 the 12th International Workshop on Computer Science and Engineering*, WCSE, 2022. doi: 10.18178/wcse.2022.06.043.

[8] T. Zebin, S. Rezvy, and Y. Luo, "An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2339–2349, 2022, doi: 10.1109/TIFS.2022.3183390.

[9] A. Almusawi and H. Amintoosi, "DNS tunneling detection method based on multilabel support vector machine," *Security and Communication Networks*, vol. 2018, pp. 1–9, 2018, doi: 10.1155/2018/6137098.
[10] D. Vekshin, K. Hynek, and T. Cejka, "DoH Insight: Detecting DoH by machine learning," in *ACM International Conference Proceeding Series*, New York, NY, USA: ACM, Aug. 2020, pp. 1–8. doi: 10.1145/3407023.3409192.
[11] Y. M. Banadaki, "Detecting malicious DoH traffic in domain name system using machine learning classifiers," *Journal of Computer Sciences and Applications*, vol. 8, no. 2, pp. 46–55, Aug. 2020, doi: 10.12691/jcsa-8-2-2.
[12] S. K. Singh and P. K. Roy, "Detecting malicious DoH traffic using machine learning," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, IEEE, Dec. 2020, pp. 1–6. doi: 10.1109/3ICT51146.2020.9312004.
[13] J. Fesl, M. Konopa, and J. Jelínek, "A novel deep-learning based approach to DoH network traffic detection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, p. 6691, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6691-6700.
[14] Y. Li, A. Dandoush, and J. Liu, "Evaluation and optimization of learning-based DoH traffic classification," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ISNCC52172.2021.9615659.
[15] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH tunnels using time-series classification of encrypted traffic," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, Aug. 2020, pp. 63–70. doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.
[16] Q. Abu Al-Haija, M. Alohaly, and A. Odeh, "A lightweight double-stage scheme to identify malicious DoH traffic using a hybrid learning approach," *Sensors*, vol. 23, no. 7, p. 3489, Mar. 2023, doi: 10.3390/s23073489.
[17] R. Mitsuhashi, A. Satoh, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Identifying malicious DNS tunnel tools from DoH traffic using hierarchical machine learning classification," 2021, pp. 238–256. doi: 10.1007/978-3-030-91356-4_13.
[18] N. Japkowicz and M. Shah, "Evaluating learning algorithms: a classification perspective," *Evaluating Learning Algorithms: A Classification Perspective*, vol. 9780521196, pp. 1–406, 2011, doi: 10.1017/CBO9780511921803.
[19] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
[20] "Scikit-learn: machine learning in Python — scikit-learn 1.4.1 documentation." https://scikit-learn.org/stable (accessed Mar. 21, 2024).
[21] "TensorFlow," *TensorFlow*. www.tensorflow.org/ (accessed Mar. 21, 2024).
[22] "Scapy," *CC-BY-SA-2.5*. https://scapy.net/ (accessed Mar. 21, 2024).
[23] R. Rodríguez, "Introduction to Mallory Proxy," *Security Art Work*.
[24] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982, doi: 10.1148/radiology.143.1.7063747.
[25] A. C. J. W. Janssens and F. K. Martens, "Reflection on modern methods: Revisiting the area under the ROC curve," *International Journal of Epidemiology*, vol. 49, no. 4, pp. 1397–1403, Aug. 2020, doi: 10.1093/ije/dyz274.

BIOGRAPHIES OF AUTHORS

Abid Dhiya Eddine is a master's holder in artificial intelligence and decision making from the Tahri Mohamed University of Bechar, Faculty of Exact Sciences and currently. He is a Ph.D student at the same university. His broad research interests cover topics relating to cybersecurity, artificial intelligence, and softwares engineering. He can be contacted at email: abid.dhiyaeddine@gmail.com.

Ghazli Abdelkader is a Ph. D in Computer Science. He received the diploma of teaching in Computer Science from the University of Bechar, University of Science and Technology USTO of Oran, Algeria in 2009. He is a Lecturer at the University of Tahri Mohamed of Bechar, Algeria, His research interests are cryptography and security. He can be contacted at email: ghazek@gmail.com.